UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/807,833	03/24/2004	Theodore C. Goldstein	2095.001000/P3125US1	4323

23720 7590 04/16/2008
WILLIAMS, MORGAN & AMERSON
10333 RICHMOND, SUITE 1100
HOUSTON, TX 77042

EXAMINER	
WU, JUNCHUN	
ART UNIT	PAPER NUMBER
2191	
MAIL DATE	DELIVERY MODE
04/16/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/807,833	GOLDSTEIN ET AL.	
	Examiner	Art Unit	
	JUNCHUN WU	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 January 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-53* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-53* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _____ is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-53 are pending in this application.

2. This office action is in response to the amendment filed on Jan. 21, 2008.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims are 1, 2, 4-8, 10-11, 18, 20-23, 28, 30-33, and 38-43 are rejected under 35 U.S.C. 102(e) as being anticipated by Evans et al. (US Pat. No. 6, 836,884 B1, hereinafter "Evans").

Per claim 1

Evans discloses

- A method for use in developing a program, comprising compiling at least a portion of a source code program defined by a waypoint during the editing of the source code program (col.2 lines 38-52 e.g. "*compiling the edited intermediate language component using the intermediate language (e.g., JIT) compiler to create an edited native code component, which may then be executed from the point where program execution left off*". & *"The invention thus advantageously allows a software developer to interactively execute portions of the code, make revisions or changes, and continue execution without having to restart the program execution from the beginning after each edit"*).

Per claim 2

the rejection of claim 1 is incorporated and Evans further discloses

- identifying the waypoint in an edited source code during editing of the source code (col.2 lines 22-25 "*The invention provides for partial execution of a native program in common language runtime system via an edit and continue component, wherein execution may be suspended at a point in the program.*") .
- compiling the source code up to the identified waypoint before completing the edit of the source code (col.2 lines 31-34 "*While program execution is suspended, the user may modify or edit one or more portions of the source code component, and resume execution of the program at the point where execution was suspended.*").

Per claims 4 and 10

the rejection of claims 1 and 8 are incorporated and Evans further discloses

- identifying a second waypoint in the source code during editing of the source code; and compiling the source code from the first waypoint to the second waypoint before completing editing of the source code (col.5 lines 43-47 *"user allow to edit the source code component which is compiled ..."*).

Per claim 5

the rejection of claim 1 is incorporated and Evans further discloses

- completing editing of the source code; and compiling the source code from the second waypoint to the end of the source code (col.5 lines 50-55 *"...executing the edited native code component ..."*).

Per claim 6

the rejection of claim 1 is incorporated and Evans further discloses

- saving the edited source code (col.3 lines 1-5 ; editing tools which implicitly comprise save functionality).

Per claims 7 and 11

the rejection of claims 1 and 8 are incorporated and Evans further discloses

- compiling the source code from the waypoint to the end of the source code upon completing editing of the source code (col.3 lines 15-21).

Per claim 8

Evans discloses

- A method for use in developing a program, comprising: identifying a waypoint in an edited source code program during editing of the source code program (col.2 lines 22-25 "*The invention provides for partial execution of a native program in common language runtime system via an edit and continue component, wherein execution may be suspended at a point in the program.*"); and compiling the source code program up to the identified waypoint before completing editing of the source code program (col.2 lines 31-34 "*While*

program execution is suspended, the user may modify or edit one or more portions of the source code component, and resume execution of the program at the point where execution was suspended.").

Per claim 18

Evans discloses

A method for suspending compiler execution prior to reaching the end of a source code program, comprising:

- identifying a waypoint in the source code program (col.2 lines 22-25 "*The invention provides for partial execution of a native program in common language runtime system via an edit and continue component, wherein execution may be suspended at a point in the program.*").
- compiling a portion of the source code program whose lower bound is defined by the identified waypoint (col.2 lines 45-52 *"In addition, the as-needed compilation avoids unnecessary compiling where an edited portion of the program, such as a method, is not called subsequent to the point where execution is restarted. The invention thus advantageously allows a software developer to interactively execute portions of the code, make revisions or changes, and continue execution without having to restart the program execution from the beginning after each edit."*)
- suspending compilation of the source code program once the portion whose lower bound is identified by the waypoint is compiled (col.2 lines 31-34 "*While program execution is suspended, the user may modify or edit one or more portions of the source code*

component, and resume execution of the program at the point where execution was suspended.").

Per claim 20

the rejection of claim 18 is incorporated and Evans further discloses

- suspending compilation of the source code program once the portion whose lower bound is identified by the waypoint is compiled includes at least one of removing a corresponding task from a work queue in an IDE, storing the compiled code in a shadow location, and suppressing errors or warning (col.7 lines 50-53 *"...executing a first portion of a native code component and suspending execution of the native code component at a first point"*)..

Per claim 21

the rejection of claim 18 is incorporated and Evans further discloses

- the upper bound of the portion is defined by the start of the source code program or another waypoint (col.2 lines 46-53).

Per claim 22

Evans further discloses

A method for resuming compiler execution of a suspended compilation, comprising:

- triggering the compilation of a portion of a source code program whose upper bound is defined by an identified waypoint and compiling the portion of the source code program

whose upper bound is defined by the identified waypoint (col.7 lines 50-64 *"method comprises executing a first portion of native code..."*).

Per claim 23

the rejection of claim 22 is incorporated and Evans further discloses

- triggering the compilation of the portion of the source code includes identifying the waypoint (col.2 lines 22-25 "*The invention provides for partial execution of a native program in common language runtime system via an edit and continue component, wherein execution may be suspended at a point in the program.*").

Per claim 28

Evans further discloses

A method for building a source code program capable of suspending and resuming compilation, comprising:

- identifying a waypoint in a source code program being edited (col.2 lines 22-25 "*The invention provides for partial execution of a native program in common language runtime system via an edit and continue component, wherein execution may be suspended at a point in the program.*").
- triggering a compilation of a portion of the source code program defined by the waypoint and compiling the portion of the source code program defined by the waypoint (col.7 lines 50-64 *"method comprises executing a first portion of native code..."*).

- suspending the compilation of the portion defined by the waypoint once the compilation reaches the waypoint (col.7 lines 50-53 *"...executing a first portion of a native code component and suspending execution of the native code component at a first point"*).
- triggering the compilation of the remainder of the source code program; and resuming the compilation of the source code program to compile the remainder (col.2 lines 34-41 *"resume execution of the program at the point where execution was suspended..."*).

Per claim 30

the rejection of claim 28 is incorporated and Evans further discloses

- triggering the compilation of the portion of the source code includes identifying the waypoint (col.7 lines 50-64 *"method comprises executing a first portion of native code..."*).

Per claim 31

the rejection of claim 28 is incorporated and Evans further discloses

- suspending compilation of the source code program once the portion whose lower bound is identified by the waypoint is compiled includes at least one of removing a corresponding task from a work queue in an IDE, storing the compiled code in a shadow location, and suppressing errors or warning (col.7 lines 50-53 *"...executing a first portion of a native code component and suspending execution of the native code component at a first point"*)..

Per claim 32

the rejection of claim 28 is incorporated and Evans further discloses

- the upper bound of the portion is defined by the start of the source code program or another waypoint (col.2 lines 46-53).

Per claim 33

the rejection of claim 28 is incorporated and Evans further discloses

- triggering the compilation of the remainder of the source code program includes identifying a second waypoint, saving the source code program, or ending an editing session (col.2 lines 34-41 *"resume execution of the program at the point where execution was suspended..."*).

Per claim 38

Evans discloses

A method for managing the output of a compile, comprising:

- compiling at least a portion of a source code program defined by a waypoint during the editing of the source code program in a first phase (col.2 lines 31-34 "*While program execution is suspended, the user may modify or edit one or more portions of the source code component, and resume execution of the program at the point where execution was suspended.*").

- compiling the remainder of the source code program in a subsequent phase (col.2 lines 34-41 *"resume execution of the program at the point where execution was suspended..."*).
- notifying a user of any errors that may have occurred during the compilation (col.21 & 22 see Interface ICorDebugErrorInfoEnum in Table).

Per claim 39

the rejection of claim 38 is incorporated and Evans further discloses

- the portion comprises a portion of the source code program defined by the start of the source code program and the waypoint (col.3 lines 28-33 "*The OS module list produced by the scanner may identify both operating system modules and portions of those modules needed for the application program, the portions identified by conditional compiler directives instructing the selective compiler to compile only the portions of the modules which it receives.*")

Per claim 40

the rejection of claim 38 is incorporated and Evans further discloses

- the portion comprises a portion of the source code program defined by the waypoint and the end of the source code program (col.8 lines 18-24).

Per claim 42

the rejection of claim 38 is incorporated and Evan further discloses

- comprising scrapping the compiled first and second portions (col.8 lines 52-56).

Per claim 43

the rejection of claim 42 is incorporated and Evan further discloses

- scrapping the compiled first and second portions includes one of scrapping the compiled first and second portions responsive to the notification and scrapping the compiled first and second portions responsive to a user input (col.14 lines 30-35).

5. Claims 12, 15-17, 24, 27, 44-53 are rejected under 35 U.S.C. 102(b) as being anticipated by Keeley (US Pat. No. 6, 138,271 B1, hereinafter "Keeley").

Per claim 12

Keeley discloses

A method for modifying a compiler to engage in rapid compilation, comprising:

- identifying a file reader portion of the compiler (col.6 lines 24-29 "*an operation designated by the name 40 "F1" might be contained at a module 44 at a defined set of instructions following a header F1. Of course it is not necessary that the operation 44 be contiguous provided its location can be defined.* ").
- modifying the identified file reader to read a portion of a source code program defined by a waypoint from a standard input (col.6 lines 36-47 *"Within each operation of the uncompiled operating system are conditional compilation instructions ... "*).

Per claim 15

the rejection of claim 12 is incorporated and Keeley further discloses

- the waypoint is identified by one of identifying the waypoint from a static definition and identifying the waypoint from a dynamic definition (col.8 lines 19-24).

Per claim 16

the rejection of claim 12 is incorporated and Keeley further discloses

- the waypoint defines a lower bound of the portion of the source code program (col.8 lines 32-36 *"the part of operation F1 that is compiled when the define flag X is defined..."*).

Per claim 17

the rejection of claim 12 is incorporated and Keeley further discloses

- the waypoint defines an upper bound of the portion of the source code program (col.8 lines 32-36 *"the part of operation F1 that is compiled when the define flag X is defined..."*).

Per claim 24

Keeley discloses

A method for identifying a command and associating it with a file that is being edited, comprising:

- modifying a file reader of a compiler to read from a standard input (col.6 lines 24-29 *"an operation designated by the name 40 "F1" might be contained at a module 44 at a*

defined set of instructions following a header F1. Of course it is not necessary that the operation 44 be contiguous provided its location can be defined. ").

- triggering the compilation of a portion of a source code program whose upper bound is defined by an identified waypoint; invoking the compiler to read the file from the modified file reader through the standard input (col.6 lines 36-47 *"Within each operation of the uncompiled operating system are conditional compilation instructions ..."*)

Per claim 27

the rejection of claim 24 is incorporated and Keeley further discloses

- triggering the compilation of the portion of the source code includes identifying the waypoint (col.6 lines 36-47 *"Within each operation of the uncompiled operating system are conditional compilation instructions ..."*).

Per claim 44

Keeley discloses

A method for use in developing a program, comprising:

- identifying at least two or more instructions in a file to compile (col.6 lines 36-40).
- compiling the identified instructions while the file is being edited (col.6 lines 41-45).

Per claim 45

the rejection of claim 44 is incorporated and Keeley further discloses

- The instructions are identified at a predetermined line number in the source code program, identifying the instructions at the point of insertion for a text editor, identifying the instructions after a predetermined number of branches as conditionals, identifying the instructions at a predetermined text offset (col.18 lines 22-27)

Per claim 46

the rejection of claim 44 is incorporated and Keeley further discloses

- identifying at least two more instructions in the file during editing; and compiling the second two or more instruction while the file is being edited (col.6 lines 25-26 *"F1 contained a module at a defined set of instructions following a header"*).

Per claim 47

the rejection of claim 44 is incorporated and Keeley further discloses

- completing editing of the file; and compiling the remainder of the edited file (col.8 lines 25-30).

Per claim 48

the rejection of claim 44 is incorporated and Keeley further discloses

- comprising saving the edited file (col.2 lines 52-53; editing tools which implicitly comprise save functionality).

Per claim 49

the rejection of claim 44 is incorporated and Keeley further discloses

- compiling the remainder of the edited file upon completing editing of the file (col.8 lines 25-30).

Per claim 50

Keeley discloses

A method for compiling a source code program, comprising:

- identifying an upper bound for a portion of the source code program to compile (col.6 lines 36-40)
- identifying a lower bound for the portion (col.6 lines 36-40)
- compiling the portion defined by the upper and lower bounds during an editing session on the source code program (col.16 lines 41-47 *"conditional compilations instructions ... are conditional on define flag ..."*).

Per claim 51

the rejection of claim 50 is incorporated and Keeley further discloses

- at least one of identifying the upper bound and identifying the lower bound includes one of identifying the bound from a static definition and identifying the bound from a dynamic definition (col.6 lines 21-29 *"Refer to Fig. 4 computer code comprising each module 44 of the operating system 18" is structured to be contained in a well-defined location held in an operating system index 38. For example an operation designated by*

the name 40 "F1" might be contained at a module 44 at a defined set of instructions following a header F1.").

Per claim 52

the rejection of claim 50 is incorporated and Keeley further discloses

- identifying a third bound in the edited source code during editing of the source code; and compiling the source code from the lower bound to the third bound before completing editing of the source code (col.8 lines 32-39).

Per claim 53

the rejection of claim 50 is incorporated and Keeley further discloses

- compiling the source code from the lower bound to the end of the source code upon completing editing of the source code (col.8 lines 18-22).

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 3, 9, 19, 29 and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Evans, in view of Keeley.

Per claims 3 and 9

the rejection of claims 1 and 8 are incorporated

Evans does not implicitly discloses

- identifying the waypoint includes one of identifying the waypoint from a static definition and identifying the waypoint from a dynamic definition

But Keeley discloses

- identifying the waypoint includes one of identifying the waypoint from a static definition (col.6 lines 25-28 *"F1 contained at a module at a defined set of instructions following a header"*) and identifying the waypoint from a dynamic definition (col.6 lines 36-40 *"...conditional compilation instructions of the form ..."*).
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify teaching of Evans with the teachings of Keeley to include identifying the waypoint includes one of identifying the waypoint from a static definition and identifying the waypoint from a dynamic definition in order to provide a method for the application program where portion identified by conditional compiler directives instructing the selective compiler to compile only portions of the modules it receives (col.3 lines 27-33).

Per claim 19

the rejection of claim 18 is incorporated

reject the same reason as claim 3

Per claim 29

the rejection of claim 28 is incorporated

reject the same reason as claim 3

Per claim 41

the rejection of claim 38 is incorporated

reject the same reason as claim 3

8. Claims 13, 14, 25, and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Evans, in view of Meth (U.S. Pub No. 20020087916 A1).

Per claims 13 and 25

the rejection of claim 12 is incorporated

But Evans does not include

- modifying the identified file reader to read from the standard input includes modifying the identified file reader to read from an open system call.

However Meth discloses

- modifying the identified file reader to read from the standard input includes modifying the identified file reader to read from an open system call ([0039] *"whenever the program opens a file with the open( ) system call, the Condor user-level checkpoint mechanism intercepts the open() system call and records for itself the name of the file being opened"*).

- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify teaching of Evans with the teachings of Meth to include modifying the identified file reader to read from the standard input includes modifying the identified file reader to read from an open system call in order to use to open and close files whenever the program opens a file with the open system call, the user-level checkpoint mechanism intercepts the open system call and records for itself the name of the file being open (see [0039]).

Per claims 14 and 26

the rejection of claim 13 is incorporated and Meth further discloses

- modifying the identified file reader to read from the open system call includes modifying the identified file reader to read from a UNIX gcc command [0039] *"... via standard UNIX system calls..."* which is implicitly included gcc command).

9. Claims 34-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Evans, in view of Sollich (U.S. Pub No. 20020016953 A1).

Per claim 34

Evans discloses

- A method for using a UNIX standard input read mechanism for speculative compilation of a source code program, comprising: identifying a waypoint in an edited source code program during editing of the source code program (col.2 lines 22-25 "*The invention*

provides for partial execution of a native program in common language runtime system via an edit and continue component, wherein execution may be suspended at a point in the program.").

But Evans does not disclose

- invoking a compile of at least a portion of a source code program defined by a waypoint during the editing of the source code program with a UNIX input read mechanism.

However Sollich discloses

- invoking a compile of at least a portion of a source code program defined by a waypoint during the editing of the source code program with a UNIX input read mechanism (Sollich [0061] *"...the Integrated Development Environment or IDE invokes the compiler for determining an appropriate context for the source code, based on where the screen cursor is currently positioned within the code."*
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify teaching of Evans with the teachings of Sollich to include invoking a compile of at least a portion of a source code program defined by a waypoint during the editing of the source code program with a UNIX input read mechanism in order to get a result to the IDE which describes the current context within the source code from compiler (see [0061]).

Per claim 35

the rejection of claim 34 is incorporated and Evans further discloses

- the portion comprises a portion of the source code program defined by the start of the source code program and the waypoint (col.2 lines 53-60).

Per claim 36

the rejection of claim 34 is incorporated and Evans further discloses

- the portion comprises a portion of the source code program defined by the waypoint and the end of the source code program (col.2 lines 53-62).

10. Claim 37 is rejected under 35 U.S.C. 103(a) as being unpatentable over Evans, in view of Sollich and further view of Keeley.

Per claim 37

the rejection of claim 34 is incorporated

Both Evans and Sollich do not disclose

- the waypoint is identified by one of identifying the waypoint from a static definition and identifying the waypoint from a dynamic definition

But Keeley discloses

- the waypoint is identified by one of identifying the waypoint from a static definition (col.6 lines 25-28 *"F1 contained at a module at a defined set of instructions following a header"*) and identifying the waypoint from a dynamic definition (col.6 lines 36-40 *"...conditional compilation instructions of the form ..."*).
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine teachings of Evans and Sollich and further include

the waypoint is identified by one of identifying the waypoint from a static definition and identifying the waypoint from a dynamic definition by the teachings of Keeley in order to provide a method for the application program where portion identified by conditional compiler directives instructing the selective compiler to compile only portions of the modules it receives (col.3 lines 27-33).

Response to Arguments

Applicant's arguments filed on Jan. 21, 2008 have been fully considered but they are not persuasive.

- In the remarks, Applicant argues that:

(a) In regard to independent claims 1, 12, 18, 22, 24, 28, 38, 44 and 50 recites a "waypoint" and performing some act defined relative to such a waypoint and each of the dependent claims 2-11, 15-17, 19-21 23, 27, 29-33, 39-43, 45-49, and 51-53 incorporates this limitation. Applicant respectfully submits those claims are novel.

Examiner's response:

Examiner disagrees.

(a) Applicant's arguments with respect to those claims above have been considered but are moot in view of the new ground(s) of rejection - see Evans et al. and Keeley arts made of record, as applied hereto, when considered alone or in combination.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Junchun Wu whose telephone number is 571-270-1250. The examiner can normally be reached on 8:00-17:00 M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JW

/Wei Zhen/

Supervisory Patent Examiner, Art Unit 2191